

# **RANCANG BANGUN PURWARUPA SISTEM PENDETEKSI KENDARAAN MENGUNAKAN PUSTAKA OPENCV**

<sup>1</sup>M. Dwiyanto Tobi Sogen

<sup>2</sup>Tubagus Maulana Kusuma

<sup>1</sup>Teknik Elektro, Politeknik Katolik Saint Paul, dwiyanto.25@gmail.com

<sup>2</sup>Fak. Teknologi dan Rekayasa, Univ. Gunadarma, mkusuma@staff.gunadarma.ac.id

## **ABSTRAK**

Video pemantau lalu lintas pada umumnya tidak mendeteksi kecepatan, jumlah, dan ukuran panjang kendaraan yang melintas. Penelitian ini berisi perancangan, pembuatan, dan analisis sistem pendeteksi kendaraan dengan teknologi *computer vision* menggunakan pustaka OpenCV. Sistem ini dapat menampilkan output berupa jumlah, kecepatan, dan ukuran panjang kendaraan. Penelitian dilakukan menggunakan OpenCV sebagai *library* pemrograman bahasa C++. Pada penelitian ini alat yang digunakan adalah kamera *web* yang dipasang di jembatan penyeberangan. Kamera *web* akan merekam pergerakan kendaraan pada jarak tertentu. Untuk mendeteksi pergerakan kendaraan maka diperlukan suatu citra latar belakang. Citra latar belakang diperoleh dengan menggunakan metode *Time Average Background Image* (TABI). Pengurangan *pixel* antara citra latar belakang dengan citra yang ditangkap oleh kamera *web* akan menghasilkan kendaraan yang terdeteksi. Kendaraan tersebut akan ditandai dengan kotak warna merah. Kecepatan suatu kendaraan dapat diukur dengan membagi jarak tempuh kendaraan dengan waktu tempuh kendaraan. Sistem ini telah berhasil menghitung kecepatan mobil yang melintas dengan kisaran galat sekitar 2,44 sampai 4 persen. Jumlah kendaraan juga dapat dihitung dengan galat sebesar 26,7 persen, dan ukuran panjang kendaraan juga dapat dihitung dengan galat sebesar 11%.

Kata kunci : citra, kamera *web*, opencv

## **ABSTRACT**

Video monitors traffic generally does not detect the speed, quantity, and length of passing vehicles. This research provides the design, manufacture, and analysis of vehicle detection system with computer vision technology using OpenCV library. The system can display the output of the number, speed, and length of the vehicle. The study was conducted using OpenCV as a library of C++ programming language. In this research tool used is a web camera mounted on the pedestrian bridge. Web cameras will record the movement of the vehicle at a certain distance. To detect the movement of vehicles, we need a background image. Background image was obtained by using the method of *Time Average Background Image* (TABI). Reduction of pixels between the background image with the image captured by the web camera will produce a vehicle is detected. The vehicles will be marked with a red box. The speed of a vehicle can be measured by dividing the distance traveled by the vehicle travel time of vehicles. This

system has been successfully calculate the speed of a passing car with an error range of about 2.44 to 4 percent. The number of vehicles can also be calculated with an error of 26.7 percent, and the length of the vehicle can also be calculated with an error of 11%.

Keywords : images, webcamera, opencv

## PENDAHULUAN

Kota Sorong adalah sebuah kota di Provinsi Papua Barat, Indonesia. Kota ini dikenal dengan sebutan Kota Minyak, di mana *Nederlands Nieuw-Guinea Petroleum Maatschappij* (NNGPM) mulai melakukan aktivitas pengeboran minyak bumi di Sorong sejak tahun 1935. Seiring dengan perkembangan jaman dan di dukung oleh UU Otonomi Khusus No. 21 Tahun 2001 di Tanah Papua, Kota Sorong dengan Visinya Menjadi Kota Termaju di Kawasan Tanah Papua, telah menjadi kota dengan Pusat Bisnis, Jasa, Pertambangan dan Industri di Tanah Papua. Kota Sorong juga merupakan pintu masuk ke Tanah Papua. Dengan semakin berkembangnya Kota Sorong, juga diikuti dengan semakin meningkatnya taraf pendapatan masyarakat. Hal ini dapat dilihat dari semakin banyaknya penduduk yang memiliki kendaraan roda empat dan dua.

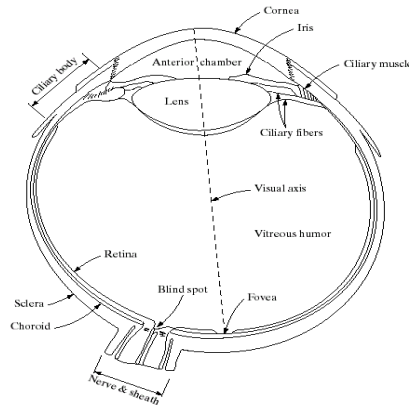
Sebagai Kota dengan kemajuan yang sangat pesat, disisi lain Pemerintah Daerah terlihat kurang memperhatikan tentang sektor infrastruktur terutama infrastruktur jalan. Sebagaimana termaktub dalam UU No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan dimana fungsi jalan adalah Aman, selamat, tertib, lancar, dan terpadu dengan moda angkutan lain, serta mendorong perekonomian nasional, dan memajukan kesejahteraan umum (Bapeda Papua Barat, 2008). Namun saat ini perkembangan volume jalan dan kapasitas jalan dan kondisi jalan Poros Kota Sorong tidak sebanding dengan meningkatnya jumlah kendaraan yang ada. Padahal infrastruktur Jalan merupakan salah satu faktor utama penggerak perekonomian, karena jalan poros ini memiliki fungsi yang signifikan yakni merupakan jalan penghubung Kota Sorong dengan kota-kota lain, jalan dengan tingkat pelayanan paling tinggi, simpul kegiatan transportasi bagi Papua Barat, pusat kegiatan jasa dan perdagangan, pusat kegiatan nasional (PKN) di Papua Barat, serta jalan poros pintu gerbang Papua Barat dan penggerak perekonomian Papua Barat. Dengan semakin minimnya infrastruktur jalan serta kondisi jalan yang rusak, masyarakat Kota Sorong mulai mengeluhkan masalah kemacetan yang melanda jalan-jalan poros kota, khususnya pada waktu pagi dan sore hari. Faktor utama yang menyebabkan kemacetan di kota sorong yakni selain jumlah kendaraan dengan kapasitas jalanan yang tidak imbang, namun dilain hal faktor terpenting adalah banyaknya kondisi jalan yang rusak. Kondisi ini disebabkan karena Pemerintah Daerah tidak memiliki suatu metode untuk memperhitungkan beban jalan dan penggunaan jalan secara efisien.

Berdasarkan hal-hal diatas, akan di lakukan sebuah penelitian pendeteksi kendaraan untuk menentukan *volume traffic* kendaraan, kecepatan kendaraan dan ukuran panjang kendaraan menggunakan *computer vision*, yang nantinya hasil dari penelitian ini diharapkan dapat membantu Pemda Kota Sorong untuk mengembangkan suatu sistem penentuan *volume traffic* kendaraan yang nantinya akan menjadi suatu kebijakan dalam menentukan perencanaan kapasitas jalan dan pengembangan infrastruktur jalan di Kota Sorong.

## TINJAUAN PUSTAKA

### *Sistem Visual Manusia*

Cahaya merupakan radiasi elektromagnetik yang menstimulasi sistem penglihatan manusia. Cahaya ditampilkan sebagai sebuah distribusi energy spectrum  $L(\lambda)$  dimana  $\lambda$  adalah panjang gelombang cahaya yang dapat diterima oleh sistem penglihatan manusia. Pembentukan citra oleh sensor mata dapat digambarkan sebagai berikut :



Gambar 1. Penampang Mata

Sumber : Gonzalez (2004)

Intensitas cahaya ditangkap oleh diagram iris dan diteruskan ke bagian retina mata. Bayangan obyek pada retina mata dibentuk dengan mengikuti konsep sistem optik dimana fokus lensa terletak antara retina dan lensa mata. Mata dan syaraf otak dapat menginterpretasi bayangan yang merupakan obyek pada posisi terbalik.

Retina mata manusia memiliki dua tipe penerima cahaya yang dinamakan *rod* dan *cone*. *Rod* berperan dalam mendapatkan citra keabuan. *Cone* banyak terdapat di sekitar retina (*fovea*). *Fovea* di bagian *retina* terdiri dari dua jenis *receptor*: Sejumlah *cone receptor*, sensitif terhadap warna, *visi cone* disebut *photopic vision* atau *bright light vision*. Sejumlah *rod receptor*, yang memberikan gambar keseluruhan pandangan dan sensitif terhadap iluminasi tingkat rendah, *visi rod* disebut *scotopic vision* atau *dim-light vision*.

### **Citra**

Citra (*image*) adalah suatu representasi, kemiripan atau imitasi dari suatu objek atau benda. Citra dapat direpresentasikan dengan dua cara yaitu citra analog dan citra digital. Pada penelitian ini citra yang digunakan adalah citra digital. Citra analog adalah citra yang bersifat kontinu. Citra analog tidak mudah direpresentasikan dalam komputer sehingga tidak bisa diproses secara langsung. Citra *digital* merupakan citra yang dapat diolah oleh komputer. Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut *picture element (pixel)*. *Pixel* merupakan satuan pemetaan terkecil yang terlihat dalam suatu layar monitor dan masing-masing titik dalam komputer grafik dapat diwakili oleh satu atau lebih *pixel* yang digambarkan ke bidang layar monitor (Paul, 2006). Representasi citra digital berukuran N baris dan M kolom ditunjukkan pada persamaan 1 (Ningsih, 2011).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

dengan  $f(x,y)$  menyatakan intensitas *pixel* pada koordinat  $(x,y)$ .

Ada beberapa cara untuk merepresentasikan suatu citra yaitu citra berwarna (*Red Green Blue*, RGB), keabuan (*gray*), dan biner (hitam putih). Langkah-langkah untuk memproses suatu citra adalah dengan mengubah citra berwarna menjadi citra keabuan. Citra keabuan ini selanjutnya akan diubah menjadi citra biner. Citra biner adalah citra yang hanya mempunyai dua nilai keabuan sehingga sering disebut citra hitam dan putih. Hasil citra biner dari proses pengolahan citra bisa saja tidak terlihat jelas. Supaya citra biner bisa terlihat lebih jelas maka perlu dilakukan proses morfologi. Proses morfologi merupakan teknik pengolahan citra yang didasarkan pada bentuk segmen atau *region* dalam suatu citra. Dalam proses morfologi citra diperlukan suatu struktur elemen (*strel*). Struktur elemen merupakan himpunan *sub-image* kecil yang digunakan untuk meneliti citra dalam pembelajaran propertinya. Pada penelitian ini akan dilakukan proses morfologi citra dilasi,erosi, dan *connected component* (Gonzalez, 2004).

### **Background Reconstruction**

*Background* adalah bagian gambar yang tidak banyak berubah dari serangkaian gambar bergerak. *Background* dapat diperoleh dengan cara mengambil nilai rata-rata dari serangkaian gambar, nilai rata-rata tersebut akan mendekati gambar latar belakang yang diinginkan (Hartoto, 2011).

### **Deteksi Objek Bergerak Menggunakan Background Subtraction**

Prinsip dasar deteksi gerakan (*motion detection*) adalah membandingkan antara dua buah citra  $f(x,y,t_1)$  dan  $f(x,y,t_2)$  sehingga menghasilkan citra baru  $r(x,y)$  yang memiliki nilai 0 (hitam) atau 1 (putih) dengan kriteria sebagai berikut (Hartoto, 2011).

$$r(x,y) = \begin{cases} 1, & \text{jika } |f(x,y,t_1) - f(x,y,t_2)| < T \\ 0, & \text{untuk nilai lainnya} \end{cases} \quad (2)$$

$T$  merupakan nilai *threshold* yang besarnya tergantung dengan kepekaan terhadap perubahan yang diinginkan. Citra hasil dari perbandingan ini akan mengandung objek yang bergerak. *Background subtraction* adalah proses untuk mendeteksi pergerakan atau perbedaan signifikan dengan citra referensi. Tujuan dari *background subtraction* adalah untuk memisahkan objek dari latar belakang sehingga gerakan dari suatu objek terdeteksi. Selisih *frame* dilakukan untuk mencari perbedaan nilai dari setiap *pixel* dari setiap citra.

### **Algoritma Time – average background image (TABI)**

Metode sederhana untuk mendapatkan *background* dengan cara rekonstruksi dari serangkaian gambar bergerak dapat dilakukan dengan menggunakan TABI. Metode ini akan menghasilkan gambar *background* yang merupakan pendekatan dengan cara mendapatkan rata-rata nilai *pixel* gambar berdasarkan serangkaian *frame* dari serangkaian gambar. Maka dapat diperoleh nilai rata-rata tiap *pixel* dengan menggunakan persamaan 2.4 (Hartoto, 2011).

$$\bar{F}(X,Y) = \frac{\sum_{i=1}^n f_i(x,y)}{n} \quad (3)$$

$x$  dan  $y$  merupakan baris dan kolom suatu matriks gambar

$\bar{F}(x,y)$  adalah rata-rata intensitas *pixel* pada posisi  $x,y$

$f_i(x,y)$  adalah intensitas *pixel* pada *frame*  $i$  dengan posisi  $x,y$

$n$  adalah jumlah *frame*

i adalah urutan *frame*

Langkah-langkah yang dilakukan untuk mendapatkan *foreground* dengan memisahkan dari *background* berdasarkan video dapat dijelaskan sebagai berikut (Hartoto, 2011) :

1. Ekstraksi video  
Video diekstrak untuk mendapatkan sejumlah *frame* dari video tersebut.
2. Memperoleh *frame* hasil rata-rata *frame* video.  
*Frame* hasil ekstraksi dipilih, kemudian diambil nilai rata-rata per *pixel* menggunakan persamaan 3.

### **Open Computer Vision (OpenCV)**

OpenCV adalah sebuah *library* yang berisi fungsi-fungsi pemrograman untuk teknologi *computer vision* secara *real time*. OpenCV bersifat *open source*, bebas digunakan untuk hal-hal yang bersifat akademis dan komersial. Di dalamnya terdapat *interface* untuk C++, C, Python, dan nantinya Java yang dapat berjalan pada Windows, Linux, Android, dan Mac. Terdapat lebih dari 2500 algoritma dalam OpenCV, digunakan di seluruh dunia, telah lebih dari 2.5 juta kali diunduh, dan digunakan lebih dari 40 ribu orang. Penggunaannya antara lain pada seni interaktif, inspeksi tambang, menampilkan peta di web melalui teknologi robotik. (Mostafa, 2008)

Fitur-fitur yang terdapat pada OpenCV antara lain (Gary, 2008): Manipulasi data *image* (alokasi, rilis, duplikasi, pengaturan, konversi), *Image* dan I/O video (masukan berbasis *file* dan kamera, keluaran *image/ video file*), manipulasi matriks dan vektor serta aljabar linear (produk, solusi, *eigenvalues*, SVD), beragam struktur data dinamis (daftar, baris, grafik), dasar pengolahan citra (filter, deteksi tepi, deteksi sudut, pengambilan sampel dan interpolasi, konversi warna, operasi morfologi, *histogram*), analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, transformasi Hough, perkiraan polygonal, menyesuaikan garis, delaunay *triangulation*, Kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, kalibrasi, dasar estimasi matriks, estimasi homografi, korespondensi stereo), analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran), pengenalan objek (metode *eigen*, HMM), dasar Graphical User Interface atau GUI (menampilkan *image/ video*, penanganan *mouse* dan *keyboard*, *scroll-bars*), pelabelan *image* (garis, poligon, gambar teks). Modul-modul yang terdapat pada OpenCV antara lain: *cv* – fungsi utama OpenCV, *cvaux* – fungsi penolong OpenCV, *cxcore* – pendukung struktur data dan aljabar linear, *highgui* – fungsi GUI.

### **Beberapa Penggunaan Image Processing Pada Traffic System**

Sistem pemantau *traffic cars* sebenarnya sudah pernah ditemukan oleh orang lain sebelumnya. Namun metode yang digunakan dapat berbeda-beda. Beberapa penemuan tersebut diantaranya :

#### **Sistem Pendeteksi Kemacetan (Traffic Jam Detection System) (Mostafa, 2008)**

Sistem ini bertujuan untuk mendeteksi kendaraan dan kecepatan dari kendaraan tersebut. Sistem ini menggunakan metode Blob Detection dengan latar belakang jalan yang kosong untuk mendeteksi kendaraan tersebut. Sistem ini menggunakan pemrograman .Net dan bahasa C. Secara umum tahapan yang dilakukan yaitu :

1. *Image analysis* ; menggunakan citra jalanan yang kosong

2. *Object detection* ;menggunakan metode *lane masking*, konversi ke *grayscale*, lalu gambar Blob yang menunjukkan kendaraan akan muncul
3. Hitung jumlah kendaraan
4. *Motion detection* ;mendeteksi kecepatan dari kendaraan dengan menggunakan metode *image differentiation*
5. Menampilkan hasil penghitungan

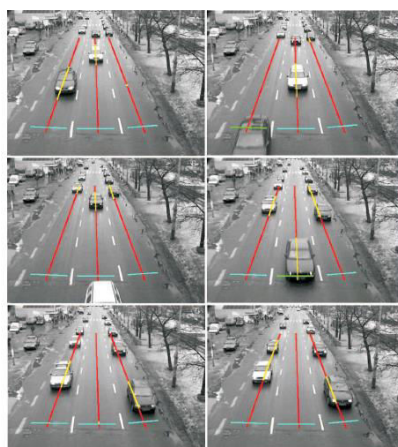
Sistem ini telah dicoba di beberapa kondisi. Hasil pengujian menunjukkan bahwa sistem ini membutuhkan waktu 3 sampai 4 detik untuk mendeteksi objek dan kecepatannya dalam citra berukuran 1024\*768 *pixels*.

### **Pemanfaatan Pengolahan Citra Pada Analisis Lalu Lintas Jalan (*Image Processing in Road Traffic Analysis*) (Atkociounas, 2005)**

Sistem ini bertujuan untuk mendeteksi kemacetan, kecepatan kendaraan, dan plat nomor kendaraan. Sistem ini menggunakan metode *lane masking*, metode *Blob Detection* menggunakan citra jalan yang kosong, dan juga algoritma *edge detection*. Secara umum tahapan yang dilakukan yaitu: *Lane masking* ;memisahkan citra kendaraan dari citra yang tidak diperlukan, jadi yang diambil hanya citra jalan saja, *Background elimination* ;memisahkan citra kendaraan dari citra jalan untuk menghilangkan *noise* yang mungkin muncul seperti ranting pohon, bayangan, hujan, dan lain sebagainya, *Noise & blob filtration* ;menghilangkan *noise* kecil yang berada di sekitar citra kendaraan, *Contour extraction* ;melakukan *edge detection* pada citra dengan algoritma yang dipakai yaitu Prewitt, Sobel, dan Laplacian Zero Crossing, *Contour labeling* ;menandai kendaraan yang telah terdeteksi, *Vehicle tracking* ;melakukan tracking kendaraan.

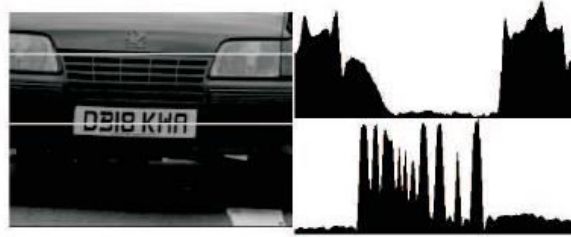
### **Penerapan Sensor Visi Untuk Pengamatan Lalu Lintas (*The Implementation of A Vision Sensor for Traffic Surveillance*) (Kasprzak, 2005)**

Sistem ini menggunakan metode *Open Model For Network-wide Heterogeneous Intersection-based Transport Management* (OMNI). Metode ini menggunakan sudut pandang jalan secara vertikal dan memanfaatkan lebar tiap lajur pada jalan untuk menghitung jumlah kendaraan seperti yang terlihat pada Gambar 2.



Gambar 2 *Display Pengujian Open Model For Network-wide Heterogeneous Intersection-based Transport Management*

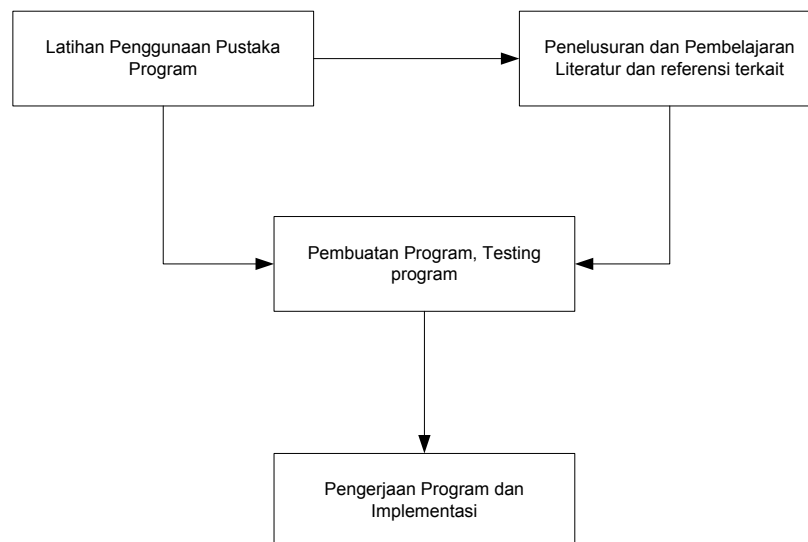
Sistem ini selain berfungsi untuk memantau kondisi lalu lintas, juga berfungsi sebagai pendeteksi plat nomor kendaraan.



Gambar 3. *Display* Pengujian Pendeteksi Plat Nomor Kendaraan

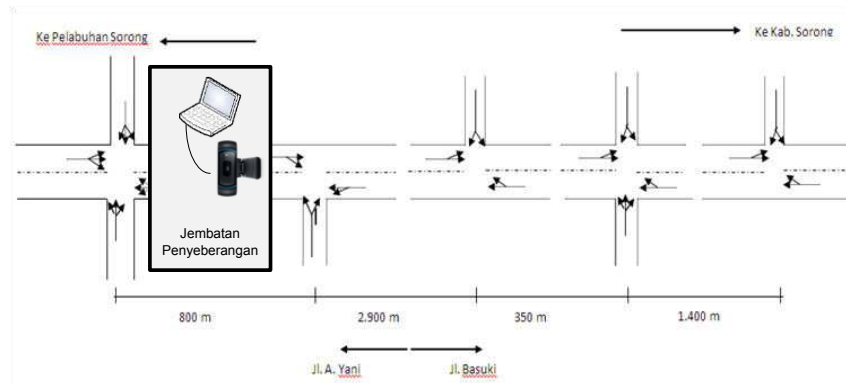
## METODE PENELITIAN

Metode penelitian yang digunakan digambarkan dalam diagram alur sebagai berikut :



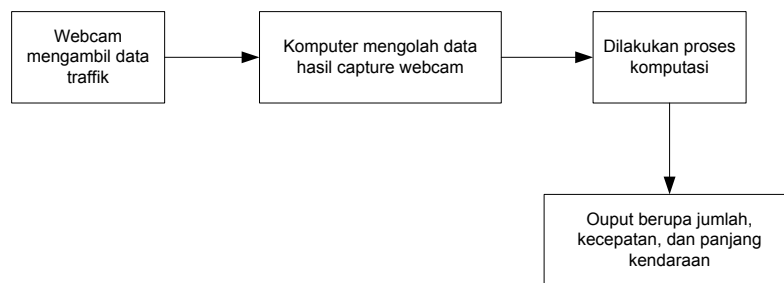
Gambar 4. Alur Penelitian

Pada perancangan sistem pendeteksi kendaraan ini, teknologi yang akan digunakan adalah *video image processing* melalui *web* kamera yang dipasang di atas jembatan penyeberangan untuk memantau jalan tersebut. Untuk pengambilan data yang akan digunakan pada pengujian, nantinya digunakan video yang berisi rekaman kondisi *traffic* kendaran pada jalan yang dijadikan objek pengamatan. Selain pengujian menggunakan *file video* hasil rekaman, juga akan dilakukan pengujian sistem secara *real time*. Pada sistem pendeteksi jumlah dan kecepatan kendaraan ini, informasi yang dibutuhkan oleh sistem adalah informasi mengenai jalan yang dipantau, khususnya mengenai jumlah kendaraan dan kecepatan kendaraan tersebut. Berikut gambaran desain sistem.



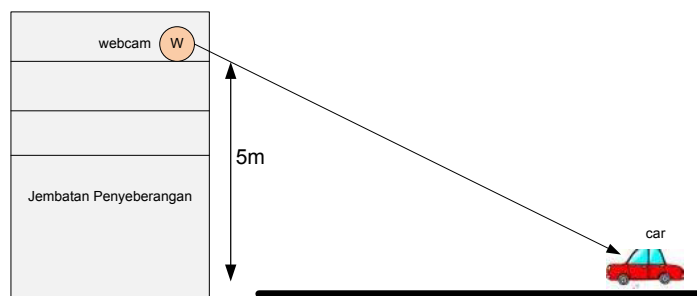
Gambar 5. Gambaran Desain Sistem

Diagram blok sistem pendeteksi kendaraan terdapat pada gambar 6 dibawah ini.



Gambar 6. Blok Diagram Sistem

Untuk peletakan *WebCam* ini (gambar 7), salah satu parameter yang perlu diperhatikan yaitu sudut pengambilan gambar. Semakin besar luas daerah yang bisa ditangkap kamera, maka semakin baik objek yang akan di deteksi. Sudut pandang dari samping atau vertical dari atas seperti dapat dianggap dapat memiliki daya tangkap yang lebih baik, namun dibutuhkan penelitian lebih lanjut mengenai dugaan tersebut.



Gambar 7. Rancangan Posisi Peletakan *Webcam*

Algoritma yang akan digunakan dalam sistem ini menggunakan bahasa Pemrograman C++. Algoritma yang akan dibuat akan mengolah data dari video yang dikirimkan oleh *web* kamera. Pengujian sistem menggunakan video yang menampilkan kondisi jalan yang sesungguhnya secara *real time* ataupun dapat berupa *file video* rekaman. Tingkat *volume* trafik yang akan dipantau menggunakan tiga buah parameter, yaitu:

1. Jumlah kendaraan.
2. Kecepatan kendaraan



3. Ukuran panjang kendaraan

Beberapa hal yang dijadikan input sistem untuk pengolahan data adalah :

- a. Video atau citra daerah yang diamati.
- b. Daerah yang akan diamati dari video atau gambar (*region of interest*).

## HASIL DAN PEMBAHASAN

Untuk mengetahui performa dari sistem ini, maka dibutuhkan pengujian dengan cara pengambilan beberapa data. Pengujian dilakukan dengan menggunakan data secara *real time* dan berupa *file video* rekaman yang berisi kondisi jalan dengan spesifikasi:

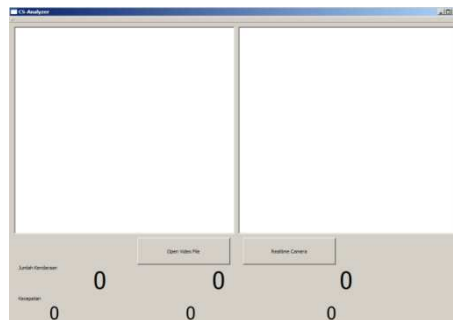
1. Durasi tiap video 1 menit
2. *Frame rate* sebesar 30 fps
3. Ukuran video 1280 x 720 *pixels*

Sedangkan mesin pengolah yang digunakan adalah sebuah komputer dengan spesifikasi:

- a. Processor Intel Core 2 Duo @1.66Ghz
- b. Memory 1536 mb
- c. VGA Card Mobile Intel 945 Express Chipset Family sebesar 256 mb

### *Pengujian Sistem Pendeteksi Kendaraan*

Panjang area jalan yang diamati adalah sekitar 15 meter. Pada pengujian ini terdapat beberapa jenis kendaraan dengan kecepatan yang berbeda-beda dan arah gerak yang dapat berubah-ubah. Hal ini dilakukan untuk menstimulasi dan menguji sistem yang dibangun apakah mampu bekerja dengan kondisi nyata dilapangan. Pada pengujian sistem penghitung jumlah kendaraan setelah tombol mulai ditekan maka kamera akan mulai menangkap *video*.



Gambar 8. Tampilan Awal Sistem Pendeteksi Kendaraan

Kode program yang digunakan untuk menangkap *video* adalah sebagai berikut:

```
CvCapture* capture = cvCreateFileCapture(videoPath.toStdString().c_str());
```

Kode program `cvCreateFileCapture(videoPath.toStdString().c_str());` berguna untuk membangun suatu koneksi antara komputer yang digunakan dengan kamera *web* yang dipasang pada suatu komputer. Selanjutnya yaitu merekonstruksi suatu latar belakang menggunakan metode TABI. Kode program untuk rekonstruksi latar belakang dapat dilihat sebagai berikut:

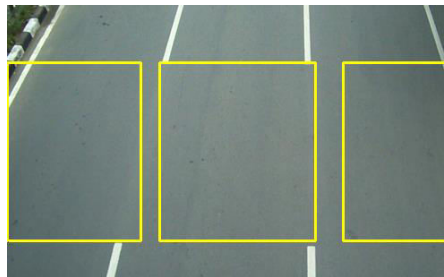
```
bg.operator()(frame, fore);
```

```
bg.getBackgroundImage(back);
```



Gambar 9. Contoh Hasil Dari Rekontruksi Latar

Setelah melakukan rekontruksi latar selanjutnya membuat daerah *Region of Interest* (ROI). Daerah ROI berguna untuk menentukan daerah pengamatan suatu kendaraan. Contoh citra dengan daerah ROI dapat ditunjukkan pada Gambar 10.



Gambar 10. Citra Dengan Daerah Ditandai (ROI)

Langkah selanjutnya citra RGB dari frame *video* tersebut akan diubah menjadi citra keabuan. Contoh perubahan dari citra RGB menjadi citra keabuan dapat ditunjukkan pada Gambar 11.



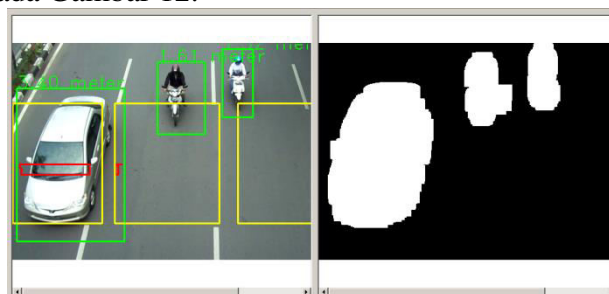
(a)



(b)

Gambar 11. Citra RGB (a), Citra Dalam Model *Grayscale* (b)

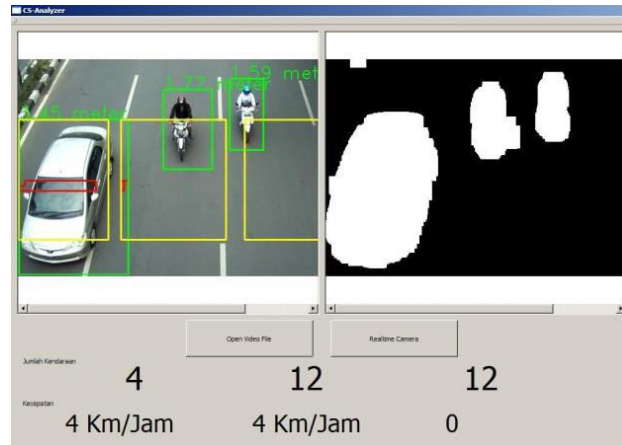
Setelah citra RGB diubah menjadi citra keabuan, maka langkah selanjutnya adalah mengubah citra keabuan menjadi citra biner. Contoh perubahan dari asli menjadi citra biner ditunjukkan pada Gambar 12.



Gambar 12. Perubahan Tampilan Citra Asli Menjadi *Binary*

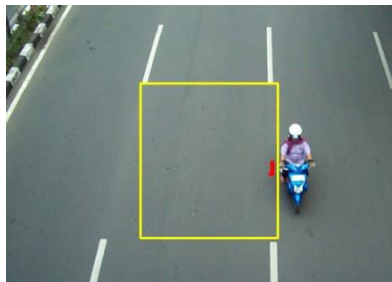
Citra biner yang didapat akan dierosi supaya menghasilkan citra yang lebih bersih. Sebelum melakukan proses erosi. Mobil yang masuk ke dalam daerah cangkupan kamera akan diberi tanda dengan kotak warna merah.

Secara keseluruhan tampilan dari sistem dapat dilihat pada gambar 13.



Gambar 13. Tampilan Dari Sistem Penghitung Jumlah, Kecepatan, dan Panjang Kendaraan

Dari pengujian dengan cara tersebut diperoleh hasil perhitungan manual sebanyak 58 kendaraan yg melewati daerah yang ditandai, sedangkan sistem menghitung sebanyak 39 kendaraan. Berdasarkan perbandingan tersebut diperoleh galat keakuratan sistem yang dibangun yaitu sebesar 26,7%. Hal ini disebabkan adanya kendaraan yang melewati begitu dekat dengan pinggir tepi daerah yang dibatasi. Hal ini ditunjukkan pada gambar 14.



Gambar 14. Tampilan Kendaraan Yang Melintas Dekat Dengan Daerah Yang Ditandai

Pada gambar 14 menunjukkan dimana sistem akan menghitung kendaraan yang melintas begitu dekat dengan daerah yang ditandai sebagai kendaraan yang termasuk dalam kategori melintasi daerah yang dibatasi, hal ini ditunjukkan dengan adanya tanda merah didekat kendaraan yang melintas tersebut. Sedangkan menurut sistem kendaraan tersebut dianggap tidak melewati daerah yang dibatasi karena badan kendaraan yang melintasi daerah yang ditandai sangat kecil.

### ***Pengujian Sistem Penghitung Kecepatan Kendaraan***

Jika kendaraan sudah melewati daerah yang ditandai (ROI) maka kecepatan kendaraan tersebut akan dihitung. Kecepatan mobil dihitung dengan cara membagi panjang daerah yang ditandai, dengan waktu kendaraan. Dalam hal ini daerah yang ditandai memiliki panjang sekitar 10 meter di keadaan yang sesungguhnya.

Beberapa pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian terhadap mobil box berwarna *orange*



Gambar 15. (a) Mobil Box Orange Memasuki ROI, (b) Mobil Meninggalkan ROI

Pada gambar 15 (a) menunjukkan gambar ketika mobil box memasuki daerah yang ditandai, sedangkan pada gambar 12 (b) terlihat mobil box telah meninggalkan daerah yang ditandai dan memunjukkan hasil kecepatan mobil box yang dihitung oleh sistem. Berdasarkan gambar 12 (b) kecepatan mobil box tersebut adalah 49 Km/jam, sedangkan berdasarkan perhitungan manual diperoleh kecepatan mobil tersebut adalah 47,4 Km/jam, dengan galat 3,33%.

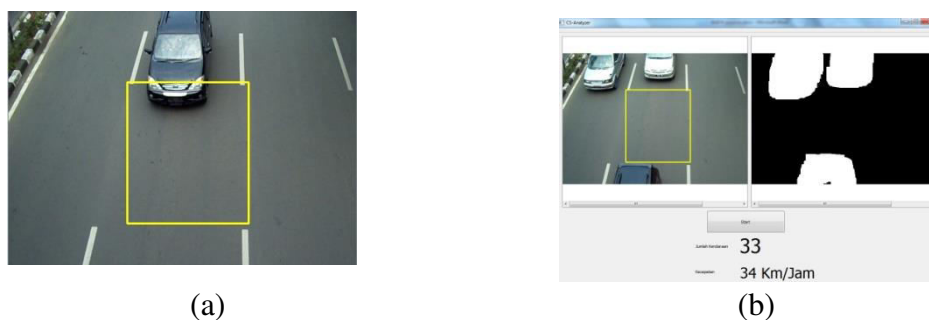
2. Pengujian terhadap truk berwarna kuning



Gambar 16. (a) Truk Kuning Memasuki ROI, (b) Truk Meninggalkan ROI

Pada mengujian terhadap truk berwarna kuning diperoleh hasil perhitungan sistem sebesar 50 Km/jam, sedangkan berdasarkan perhitungan manual diperoleh kecepatan sekitar 48 Km/jam, dengan galat 4%.

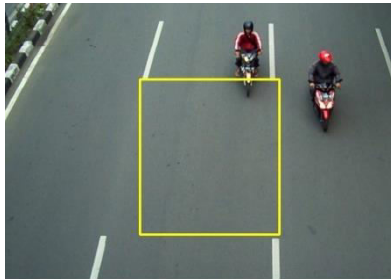
3. Pengujian terhadap mobil berwarna hitam



Gambar 17. (a) Mobil Hitam Memasuki ROI (b) Mobil Meninggalkan ROI

Pada mengujian terhadap mobil berwarna hitam diperoleh hasil perhitungan sistem sebesar 34 Km/jam, sedangkan berdasarkan perhitungan manual diperoleh kecepatan sekitar 35,3 Km/jam, dengan galat 3,81%.

#### 4. Pengujian terhadap sepeda motor



(a)



(b)

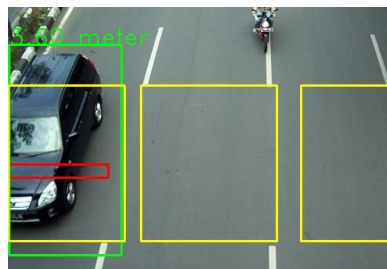
Gambar 18. (a) Motor Memasuki ROI, (b) Motor Meninggalkan ROI

Pada pengujian terhadap motor diperoleh hasil perhitungan sistem sebesar 45 Km/jam, sedangkan berdasarkan perhitungan manual diperoleh kecepatan sekitar 43,9 Km/jam, dengan galat 2,44%.

#### ***Pengujian Sistem Penghitung Ukuran Panjang Kendaraan***

Jika kendaraan sudah memasuki daerah yang ditandai (ROI) maka ukuran panjang kendaraan tersebut akan dihitung. Penghitungan ukuran kendaraan menggunakan bantuan *library opencv* yakni `cv::findContours`. Berikut tampilan hasil pengukuran panjang kendaraan :

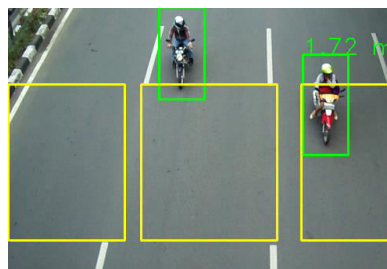
##### 1. Pengujian terhadap mobil hitam



Gambar 19. Tampilan Pengujian Ukuran Panjang Mobil Hitam

Pada pengujian terhadap mobil hitam (Nissan xtrail) diperoleh panjang sebenarnya 4 m sedangkan hasil perhitungan sistem sebesar 3,6 m, sehingga di dapat galat sebesar 11%.

##### 2. Pengujian terhadap sepeda motor



Gambar 4.15 Tampilan Pengujian Ukuran Panjang Sepeda Motor

Pada pengujian terhadap sepeda motor (Suzuki shogun) diperoleh panjang sebenarnya 1,88 m sedangkan hasil perhitungan sistem sebesar 1,72 m, sehingga di dapat galat sebesar 9,3%.

## KESIMPULAN DAN SARAN

### *Kesimpulan*

Berdasarkan hasil penelitian maka dapat diambil simpulan sebagai berikut : Perangkat lunak OpenCV digunakan sebagai *library* yang siap dipanggil dalam algoritma *computer vision* untuk melakukan komputasi penghitung jumlah, kecepatan dan ukuran panjang kendaraan. Sistem pendeteksi kendaraan yang dirancang berhasil mendeteksi jumlah kendaraan, kecepatan, dan ukuran panjang kendaraan dengan baik. Sistem ini mampu mendeteksi kecepatan kendaraan yang melintas dengan kisaran galat sekitar 2,44 sampai 4 persen. Jumlah kendaraan juga dapat dideteksi dengan kisaran galat sebesar 26,7 persen. Serta ukuran panjang kendaraan juga dapat dideteksi dengan galat sebesar 11% (untuk pengujian pada mobil), dan 9,3% (untuk pengujian pada sepeda motor). Dari hasil pengujian sistem terdapat perbedaan antara perhitungan kecepatan secara manual dan secara sistem yang tidak begitu besar selisihnya. Sehingga sistem yang diimplementasikan ini mampu menghitung kecepatan kendaraan yang bergerak dengan cukup akurat selama kendaraan tersebut tetap berada dalam lajur jalan yang ditandai.

### *Saran*

1. Sistem pendeteksi kendaraan ini diharapkan dapat diimplementasikan pada pemerintah daerah Kota Sorong dalam menganalisis pengaruh kondisi *volume* traffik jalan untuk perencanaan mutu perbaikan dan kualitas jalan, serta dapat digunakan sebagai acuan dalam analisis simpang bersinyal.
2. Sistem pendeteksi kendaraan ini diharapkan dapat digunakan sebagai acuan dalam analisis simpang bersinyal, yang dapat membantu analisa penentuan *volume* kendaraan di Kota Sorong.
3. Faktor ketinggian penempatan kamera juga sangat perlu diteliti lebih lanjut, karena akan sangat berpengaruh terhadap akurasi pendeteksian kecepatan.
4. Sistem ini diharapkan dapat dikembangkan lebih lanjut, sehingga dapat bekerja secara telemetri (pengendalian jarak jauh), maupun dapat dikembangkan dalam bentuk aplikasi *mobile*.

## DAFTAR PUSTAKA

- Atkociounas, *et al.* 2005. *Image Processing in Road Traffic Analysis*. Nonlinear Analysis: Modelling and Control, Vol. 10, No. 4, 315–332. *Proc. British Machine Vision Association Conference*
- Badan Perencanaan Pembangunan Daerah Propinsi Papua Barat, 2008, *Rencana Tata Ruang Wilayah (RTRW) Propinsi Papua Barat Tahun 2008 – 2028*
- Gonzalez, R.C., Woods, R.E., and Eddins, S.L. 2004. *Digital Image Processing Using MATLAB*. New Jersey: Pearson Prentice Hall.

- Hartoto, P. 2011. *Sistem Deteksi Kecepatan Kendaraan Bermotor Pada Real Time Traffic Information System*. Tugas akhir. Fakultas Teknik Institut Teknologi Sepuluh November, Surabaya.
- Kaehler, Adrian dan Bradski , Gary .2008. *Learning OpenCV*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472
- Khan Muhammad Nafee Mostafa & Quadrat-E-Alahy Ratul, 22 June 2008., *Traffic Jam Detection System*. CSE 3100 Report On Software Development Project-II. Khulna University of Engineering & Technology, Bangladesh.
- Ningsih, F.P. 2011. *Pengukuran Kecepatan Objek bergerak Menggunakan Webcam Berbasis Pengolahan Citra Digital*. Tugas Akhir. Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh November, Surabaya.
- Paul. 2006. *Perancangan Program Aplikasi Penghalusan dan Pembesaran Citra-JPEG Dengan Menggunakan Metoda Canny-Derliche Dan Interpolasi*. Tugas Akhir. Fakultas Teknik Informatika dan Matematika Universitas Bina Nusantara, Jakarta..
- Włodzimierz Kasprzak & Marcin Jankowski. 2005. *The Implementation of A Vision Sensor for Traffic Surveillance*. Open Model for Network-Wide-Heterogeneous Intersection- Based Transport Management (OMNI)", EC-IST 1999-11250. University of Technology, Warsaw.